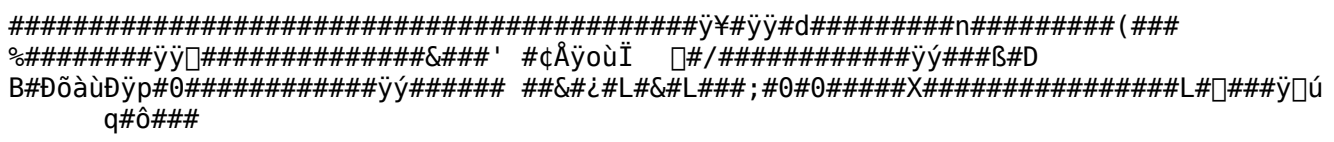

###$#############
#ÿ¥#ÿÿ#d#########n########L####úXöª#i
Vÿÿ####ÿÿ####Aÿ#ÿÿ#x#########d###########´An Information-Rich Environment in Which Explicit Connections Between All Engineering Artifacts and Their Derivatives Enable Visibility Into Extended, Distributed, Collaborative Software Engineering Processes for All Stakeholders.Navigable, query-able, meaningful information enables: informed decision-making for all stakeholdersretention of rationale and other historical informationshorter cycle times, more predictable processes#

```
###>#########################0#################>###########ÿ¥#ÿÿ#d#########d######
#####ÿ¥#ÿÿ#d#########d###########ÿ¥#ÿÿ#d#########d###########ÿ¥#ÿÿ#d#########d####
####□#□####lúj#Rþmÿÿ####ÿÿ#####ÿ□#ÿÿ#d#########d###########□SW Engr. Domain:
Problem Reports, Code, Release Descriptions, Architecture Models,Documents,
Schedules, Plans, Processes, etc.Other: metrics, n-gram1#
```

```
################$#########################ÿ¥#ÿÿ#d#########n########L#□###ûEö
¢#X#□ÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########(By-ReferenceAs 1-level Hypertext
integrate with knowledge base and provide URL orIntegrate with knowledge base and
wrap dataAs Two-Level Hypertextintegrate with knowledge base, provide data wrapper
and analyzerAs Evolver-Based Toolintegrate as two-level hypertext and provide tool
interface#
```

```
#######################################ÿ¥#ÿÿ#d#########d########Ì#□###û$ö þ#
ñÿÿ####ÿÿ#####ÿ□#ÿÿ#d#########d########### A mediator is a software module that
exploits encoded knowledge about certain sets or subsets of data to create
information for a higher layer of applications.*#
```

```
#################
########################ÿ¥#ÿÿ#d#########n########L#□###úÝöz#v#*ÿÿ####ÿÿ####Aÿ□#ÿÿ
#x#########d###########□Currently provided by RLF from WPL LabsDomain model /
class hierarchy of software engineering artifactsDifferent models for different
environments#
```

#################(#########################ÿ¥#ÿÿ#d#########n########L#□###ù□õð#¼#M ÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########□Tool integration requires data and information integration.Existing tools systems and DBs are current sources => interface to legacy systemsNew data sources need be integrated continually => extensibilityTools conventionally embed data access and data manipulation code. Applications depend on low-level details such as data values, schema, DBMS and h/w platform dependence => high life cycle costsData manipulation code which transforms data to information required by applications is typically re-implemented in each application => little reuseEnd-users and applications require information rather than access to raw data => data overload#

#################(#########################ÿ¥#ÿÿ#d#########n########L#□###ú#õ»#3
Eÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########%Interpose a specific mediation layer.
Create an information-level to insulate applications from data access dependencies
Migrate information processing code to the mediator layer.Provide structured data,
views, abstractions, aggregations in application termsIntegrate data from multiple
sources as needed and not at design timeSeparate implementation responsibilities.
Data base administrators maintain databases and data modelApplication programmers
develop end-user applicationsInformation analysts and engineers create information
mediators0#

```
###   #########   ###(############# #########
      #ÿ¥#ÿÿ#d#########n########L#□###ü#õ»#P
þÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########ìSoftware components that satisfy
requests made in shared domain terms.#Shield requestor from the (dirty) details
of:Where, how to obtain the needed information.How to process for common
abstractions, aggregations, combinations, etc..#
```

#################(#########################ÿ¥#ÿÿ#d#########n########L#□###ûxøÀ###$ÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########âSingle data source:Abstraction, aggregation and representation of data into domain relevant form.#Multiple data sources:Reconciliation of mismatches in data representation and structure.Fusion, reasoning over data sources.#

```
#################(########################ÿ¥#ÿÿ#d#########n########L#□###ùöõ»##
Eÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########Z###########□Provide API for data sources and/or
systems to enable interoperabilityDoes not interfere with original system design,
intention, requirementsÉÉbut allows other system modules/agents to access the
data/functionality of the systemApplication internals not affectedUse current
interface mechanisms or make minimal additions to provide access Existing
interfaces work as always -#wrapper is non-intrusive.#
```

#################(########################ÿ¥#ÿÿ#d#########n########L#□###ùµùÆ#Ú
-ÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########ÑWrapper usage enables:Multiple
applications to access a single resource.Via a standard interface independent of
underlying implementation (e.g., ÒCÓ today, Java tomorrow).Applications state
requests in resource terms (e.g., SQL).But suppose an application requires
multiple resources.Interpose a Mediator to do the dirty work.Determine the end
resources.Make the end requests to resources.Integrate the results.Applications
state requests in domain terms.##

```
#################(#########################ÿ¥#ÿÿ#d#########n########L#□###ú~ôè#Èþ¸
ÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d############Mediators enable...Evolvability
ReusabilityMaintainabilityScalabilityInteroperability#Technology ThrustsCreating
and sharing information  from heterogeneous data sources.Integrating legacy
systems and data.A language and protocol for information exchange.x#
```

#################(#########################ÿ¥#ÿÿ#d#########n########L#□###úçö□###R
ÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########ÙAn agent-based architecture is a collection of intelligent agents:Autonomous software modules.Actions / interactions are knowledge-driven.Communicate with other agents as needed to cooperate in accomplishing tasks.##

#################(########################ÿ¥#ÿÿ#d#########n########L#□###ûàõR##

Ûÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d############Fosters interoperation:Mosaic rather than monolithic systems.Can apply right tools for each task #(rather than force-fitting).Resource locations can be distributed.Brings combined resources to bear on problems too big / complex for single system solution.##

#################(########################ÿ¥#ÿÿ#d#########n########L#□###ûªõ»
      #
Eÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########Z###########□Agents send and/or receive messages to/from other agents. Messages contain both a communication-level and a content-level semantics.Communication semantics is expressed in an Agent Communication Language such as KQML.Content-level semantics are expressed in a language agreed upon by the agents that comprise a system, using terms for which there exist common, accepted definitions among those same agents.##

#######################################ÿ¥#ÿÿ#d#########d####### L#□###ü#õ¼#E
Eÿÿ####ÿÿ#####ÿ□#ÿÿ#d#########d###########èKnowledge Query and Manipulation Language (KQML)Developed, demonstrated and usedISO Programs: I*3, ARPI, ALP, BADD, DAISITO Programs: CAETI, EDCSKey concept:Separation of communication protocol-level and content-level languages#

```
#################(#########################ÿ¥#ÿÿ#d#########n#######
L#□###û=õð#*
zÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########d###########ÿYou must follow the same rules for
discourse in order for others to find and speak with you.Who to talk toHow to find
themHow to get their attentionHow to initiate and maintain a conversationHow to
deal with what they sayHow to respond appropriatelyÿ#
```

#########ÿ¥#ÿÿ#d#########d###########!I am not going to address that question. Because what KQML is about is the third requirement for a common language:that it have a shared pragmatics. A set of rules for meeting a person and carrying on a conversation.I am going to extend the meaning of pragmatics here, just a bit :-)!#

###0#########0###(############0#########0#ÿ¥#ÿÿ#d#########n########L####üö##`
¼ÿÿ####ÿÿ####Aÿ#ÿÿ#x#########P#2#########3A high-level communication language and protocol for exchanging information independent of content syntax and ontologyA language in which to wrap information offering a uniform view of an agentAn extensible set of performatives expressing the communication protocol to be used with the content information##

```
#################(########################ÿ¥#ÿÿ#d#########n########L#□###ùööt#%
ãÿÿ####ÿÿ#####ÿ□#ÿÿ#d#########K###########iContains a single (speech act)
performative ask        reply       forward     tell              deny
      subscribeachieve        monitor     evaluate ...With an associated semantics
and protocol With a list of attribute/value pairs that provide information used
for the communication, including;:content      :receiver   :sender:language
      :reply-with :in-reply-to:topic           :from       :to:ontology
      ...##
```

##ÿÿ#####ÿ□#ÿÿ#d#########d###########vThe Syntax for KQML is simple.It wraps the data/information in descriptive information. It is possible for a program#

```
#################(########################ÿ¥#ÿÿ#d#########n########L#□###ú-ö0#□
      rÿÿ####ÿÿ#####ÿ□#ÿÿ#x#########d###########üBasic query performatives: #
evaluate, ask-if, ask-in, ask-one, ask-all Multi-response query performatives: #
stream-in, stream-all Response performatives:#     reply, sorry Generic
informational performatives:#     tell, achieve, cancel, untell, unachieve
Generator performatives:#     standby, ready, next, rest, discard, generator
Capability-definition performatives:#     advertise, subscribe, monitor, import,
export Networking performatives: register, unregister, forward, broadcast, route#
```

##ÿÿ#####ÿ□#ÿÿ#d#########d###########RReview stream in stuff. This takes (I think) a KQML expression as a content field.#

#################(########################ÿ¥#ÿÿ#d#########n#######
L#□###ùÛ#@#`#□ÿÿ####ÿÿ####Aÿ□#ÿÿ#x#########P###########§KRILsKnowledge Router
Interface LibraryAPI for application developers for building agentsRoutersOne per
agent - handles sending and receiving of asynchronous messagesMaintains internal
cache of agent addressesRegisters and Unregisters agent with the Agent Name Server
Agent Name Server (ANS)Provides local database of agent names and addresses
Eventually a hierarchical name space, e.g., Internet Domain Name Service##